

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1-37. (canceled)

38. (currently amended) A method of providing a distributed application services integration system comprising:

providing a central domain server, wherein a configuration file resides on the central domain server, and the configuration file contains a list of cookie fields that may be read, or written to, and identifies whether a particular application has read access or write access, to a field of a cookie file;

providing a first application which transmits first application user event data to a first application interface library;

wherein the first application interface library determines whether the first application user event data is a first type of event data which requires a change to a field in a cookie file to provide real time communication to other applications of the system, and the first application interface library determines whether the user event data is a second type of user event data which does not require real time communication to other applications of the system;

using a cookie access library to update a change in a field in the cookie file where the first application user event data is determined to be a first type of user event data;

where a first application user event data is determined to be a second type of user event data transmitting the user event data as a message through a queue of a message queuing middleware; and

controlling the first applications access to cookie fields of the cookie file based on the list in the configuration file.

39. (currently amended) The method of claim 38 further including:

providing a second application which transmits a second application user event data to a second application interface library;

wherein the second application interface library determines whether the second application user event data is a first type of event data which requires a change to a field in the cookie file to provide real time communication to other applications of the system, and the second application interface library determines whether the second application user event data is a second type of user event data which does not require real time communication to other applications of the system;

using a cookie access library to update a change a field in the cookie file where the second application user event data is determined to be the first type of user event data;

where a second application user event data is determined to be the second type of user event data transmitting the second application user event data as a message through the que of the message queuing middleware; and

controlling the second applications access to cookie fields of the cookie file based on the list in the configuration file.

40. (currently amended) The method of claim 38 further including:

encrypting first application user event data where the first application user event data is determined to be the second type of event data, prior to transmitting the first application user event data as a message through the que of the message queuing middleware.

41. (previously presented) The method of claim 38 further including:

pushing information from the configuration file through the message delivering middleware to the first application interface library.

42. (previously presented) The method of claim 41, wherein the information from the configuration file controls the operation of the first application interface library when a user event takes place.

43. (previously presented) The method of claim 39 further including:
pushing information from the configuration file through the message delivering middleware to the first application interface library and the second application interface library.

44. (previously presented) The method of claim 43, wherein the information pushed from the configuration file controls the operation of the first application interface library when a user event takes place, and controls the operation of the second application interface library.